MÓDULO 4: FUNCIONES EN PHP

Eduard Lara

INDICE

1. Funciones en PHP
2. Funciones Include y Require
3. Biblioteca de funciones

1. FUNCIONES EN PHP

- Además de las funciones PHP incorporadas, podemos crear nuestras funciones.
- Una función, como en cualquier lenguaje de programación (Java, C), es un bloque de instrucciones que permite encapsular ciertas funcionalidades del código, que se puede utilizar repetidamente en un programa.

Sintaxis

```
function functionName () {
    code to be executed ;
}
```

```php
<?php
function writeMsg() {
    echo "Hello world!";
}

writeMsg(); // call the function
?>
```

1. FUNCIONES EN PHP

- Las funciones deben ser definidas antes de que puedan ser llamadas. Un lugar habitual es el head
- El formato de la cabecera de una función es:

function functionName($arg_1, $arg_2, …, $arg_n)

- Notar que el tipo de retorno no está especificado
- A diferencia de las variables, los nombres de la función no son sensibles a las mayúsculas:

foo(...) == Foo(...) == FoO(...)

1. FUNCIONES EN PHP

- Por defecto los parámetros en un función se pasan por valor:

Función	Llamada

function suma ($x, $y)
{
  $s = $x + $y;
  return $s;
}
	$a=1;
	$b=2;
	$c=suma ($a, $b);
	print $c;

1. FUNCIONES EN PHP

ANATOMIA DE UNA FUNCION

Keyword de la funció, tota funció ha de començar amb aquesta paraula "**function**"

Nom : suma

Paràmetres: $s1, $s2

```
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <?php
            function suma($s1, $s2) {
                return $s1+$s2;
            }
            echo suma(3,2);
        ?>
    </body>
</html>
```

Retorn: $s1 + $s2

Invocació: suma(3,2);

1. FUNCIONES EN PHP

ANATOMIA DE UNA FUNCION: RETORNO

Una funció pot retornar un valor o no. Si no retorna, la funció seria lo que clàssicament s'anomena **procediment**

```php
function suma($s1, $s2) {
    return $s1+$s2;
}
echo suma(3,2);
```

Retorna (lo anuncia precedido por el keyword return) en este caso el valor de la suma de las variables $s1 y $s2

```php
function suma($s1, $s2) {
    echo  $s1+$s2;
}
suma(3,2);
```

No retorna res (**procediment**) La funció mostra directament el resultat per pantalla.

1. FUNCIONES EN PHP

ANATOMIA DE UNA FUNCION: PARAMETROS

Una funció pot tenir 0 o més paràmetres.

```php
function hello_world() {
    echo "Hello World!";
}
```

Sense paràmetres

```php
function hello_world($who) {
    echo "Hello ".$who;
}

hello_world("Alumne!");
```

Amb 1 paràmetre

```php
function hello_world($name,$surname) {
    echo "Hello ".$name." ".$surname;
}

hello_world("Alumne","IT");
```

Amb 2 paràmetres

1. FUNCIONES EN PHP

ANATOMIA DE UNA FUNCION: PARAMETROS

Els paràmetres d'una funció poden tenir **valors per defecte**

```php
function hello_world($name,$surname = "IT") {
    echo "Hello ".$name." ".$surname;
}

hello_world("Alumne");
```

S'invoca la funció **sense donar valor al segon paràmetre**.

La pròpia definició de la funció, defineix un valor **per defecte** en el paràmetre **$surname**

1. FUNCIONES EN PHP

ANATOMIA DE UNA FUNCION: NOMBRE E INVOCACION

function suma($s1, $s2)

S'invoca de la mateixa manera que s'anomena la funció i amb els mateixos paràmetres,tret que n'hi hagi d'opcionals.

suma(3,2)

1. FUNCIONES EN PHP

PASO PARAMETROS POR VALOR

```
<?php
    function foo($arg_1, $arg_2) // Función
     {
        $arg_2 = $arg_1 * $arg_2;
        return $arg_2;
    }
    $result_1 = foo(12, 3);          // Almacena el resultado
    echo $result_1;           // Muestra 36
    echo foo(12, 3);          // Muestra 36
?>
```

1. FUNCIONES EN PHP

PASO PARAMETROS POR VALOR

```
23  <html>
24  <head>
25  <title>Problema</title>
26  </head>
27  <body>
28  <?php
29      function retornarpromedio($valor1,$valor2)
30      {
31        $pro=$valor1/$valor2;
32        return $pro;
33      }
34      $v1=100;
35      $v2=50;
36      $p=retornarpromedio($v1,$v2);
37      echo $p;
38      ?>
39  </body>
40  </html>
```

1. FUNCIONES EN PHP

Institut Joan d'Àustria

PASO PARAMETROS POR REFERENCIA

Las variables se pasan utilizan el símbolo '&', que indica la dirección de la variable.

Función	Llamada

function incrementa (&$a)
{
$a = $a +1;
}
	$a=1;
	incrementa ($a);
	print $a; //Muestra un 2

1. FUNCIONES EN PHP

PASO PARAMETROS POR REFERENCIA

```
42  <html>
43  <head>
44  <title>Problema</title>
45  </head>
46  <body>
47  <?php
48      function cuadradocubo($valor,&$cuad,&$cub)
49      {
50        $cuad=$valor*$valor;
51        $cub=$valor*$valor*$valor;
52      }
53      cuadradocubo(2,$c1,$c2);
54      echo "El cuadrado de 2 es:".$c1."<br>";
55      echo "El cubo de 2 es:".$c2;
56  ?>
57  </body>
58  </html>
```

1. FUNCIONES EN PHP

Institut Joan d'Àustria

PASO PARAMETROS POR DEFECTO

```
function muestranombre ($titulo = "Sr.")
{
   print "Estimado $titulo:\n";
}
muestranombre ();
muestranombre ("Prof.");
```

Salida:

```
Estimado Sr.:
Estimado Prof.:
```

1. FUNCIONES EN PHP

PASO PARAMETROS POR DEFECTO

Los argumentos con valores por defecto deben ser siempre los últimos:

```
function muestranombre ($nombre, $titulo= "Sr.")
{
   print "Estimado $titulo $nombre:\n";
}
muestranombre (“Fernández”);
muestranombre (“Fernández”, "Prof.");
```

Salida:

```
Estimado Sr. Fernández:
Estimado Prof. Fernández:
```

1. FUNCIONES EN PHP

EJEMPLO

Función que cuenta el numero de accesos a la página web

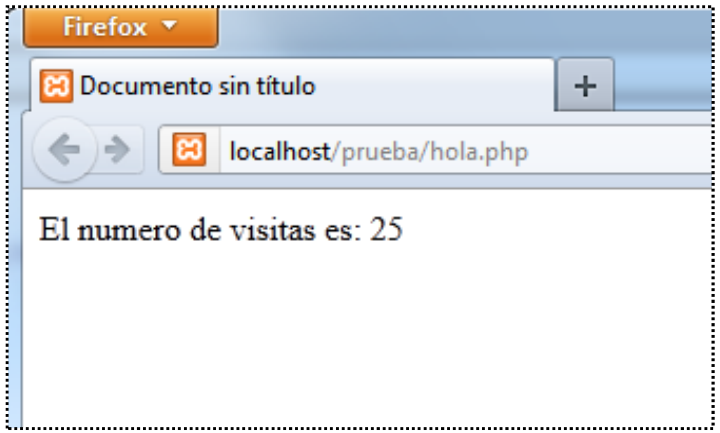
```
<?php
function contador_paginas() {
        $filename = 'webcounter.txt';        // Nuestro fichero contador
        $fp = fopen($filename,"r");          // Lo abre para lectura
        $counter = fread($fp, filesize($filename) ); // Lee el valor
        fclose( $fp );
        ++$counter;                          // Incrementa el contador
        $fp = fopen( $filename,"w");         // Lo abre para escritura
        fwrite( $fp, $counter);              // Escribe nuevo valor
        fclose( $fp );
        return $counter;
}
```

1. FUNCIONES EN PHP

EJEMPLO

En el body hacemos la llamada a la función:

```
<body>
<?php
        echo "El numero de visitas es: ".contador_paginas();
?>
</body>
```



2. FUNCIONES INCLUDE Y REQUIRE

❖ Las funciones include y require permiten la inclusión del contenido de ficheros externos dentro del código actual

❖ La diferencia está en que en caso de error include() produce un warning y require() un error fatal. S

❖ Se usará require() si al producirse un error debe interrumpirse la carga de la página

```
<?php include("ruta  fichero"); ?>
```

❖ Si se incluye código PHP, poner <?php y ?> si no sería interpretado como texto normal.

❖ Permite actualización homogénea de sitio web

2. FUNCIONES INCLUDE Y REQUIRE

```
<html>
<head><title>Fecha en castellano</title></head>
<body>
<? php
        include("fecha.inc");
        include("hora.inc");
        echo fecha ();
        echo hora () ; ?>
</body>
</html>
```

Cada una de las funciones está guardada en un archivo aparte (fecha. inc y hora. inc) y llamadas con include desde la página principal.

2. FUNCIONES INCLUDE Y REQUIRE

EJEMPLO

```
<?PHP
// Incluir bibliotecas de funciones
   require ("$libdir/conecta.php");
   require ("$libdir/fecha.php");
   require ("$libdir/cadena.php");
   require ("$libdir/globals.php");
?>

<?PHP
   include ("cabecera.html");
?>

// Código HTML + PHP
. . .

<?PHP
   include ("pie.html");
?>
```

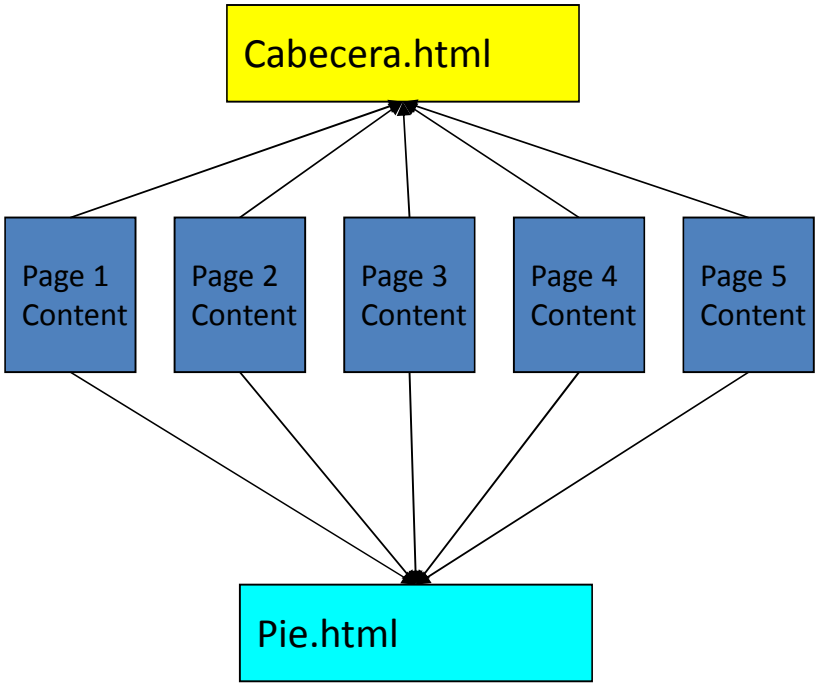
- Plantilla básica que se utilizará en todas las páginas. Cada página deberá llevar la extensión .php para que el servidor procese el codigo PHP.
- Las plantillas cabecera y pie de pagina estan en el mismo directorio.

2. FUNCIONES INCLUDE Y REQUIRE

EJEMPLO

Beneficios estructura:

- Cualquier cambio en la cabecera o en el pie de página necesita editar un único fichero. Esto reduce la cantidad de trabajo necesario para mantenimiento y rediseño.
- Ayuda a separar el contenido del diseño.



2. FUNCIONES INCLUDE Y REQUIRE

EJEMPLO

Cabecera.html

```
<html><head>
<title>UCR Webmaster Support Group</title>
<link rel="stylesheet" type="text/css" href="mycssfile.css">
</head>
<body>
<table width=80% height=30>
<tr><td>  <div align=center> Page Title </div>
</td></tr></table>
```

Pie.html

```
<table width=80% height=30>
<tr><td>
        <div align=center> UC Riverside Department<BR>
          <a href=mailto:someuser@ucr.edu>someuser@ucr.edu</a>
 </div>
</td></tr></table>
</body>
</html>
```

3. BIBLIOTECA DE FUNCIONES

- Existen muchas bibliotecas de funciones en PHP
- Algunos ejemplos:
  - Funciones de manipulación de cadenas
  - Funciones de fecha y hora
  - Funciones de arrays
  - Funciones de ficheros
  - Funciones matemáticas
  - Funciones de bases de datos
  - Funciones de red
- Algunas bibliotecas requieren la instalación de componentes adicionales
- Todas las funciones de biblioteca están comentadas en la documentación de PHP

3. BIBLIOTECA DE FUNCIONES

Institut Joan d'Àustria

Funciones de manipulación de cadenas

- explode(): Divide una cadena en subcadenas

  array **explode** (string separator, string string [, int limit])

- rtrim(), ltrim(), trim(): Eliminan caracteres a la derecha, a la izquierda o por ambos lados de una cadena.
- strstr(): Busca la 1º ocurrencia de una subcadena
- strtolower() / strtoupper(): Convierte una cadena a minúscula / mayúscula
- strcmp() / strcasecmp(): Compara dos cadenas con/sin distinción de mayúsculas

3. BIBLIOTECA DE FUNCIONES

Funciones de manipulación de cadenas

- strlen(): Calcula la longitud de una cadena
- str_split(): Convierte un string en un array
- strpbrk(): Busca una cadena por cualquiera de los elementos de un conjunto de caracteres
- substr_compare(): Comparación segura a nivel binario de 2 o más strings desde un offset hasta la longitud de caracteres

3. BIBLIOTECA DE FUNCIONES

Institut Joan d'Àustria

Funciones de arrays

- array_count_values(): Calcula la frecuencia de cada uno de los elementos de un array
- array_search(): Busca un elemento en un array
- count(): Cuenta los elementos de un array
- sort(), rsort(): Ordena y reindexa un array (r=decreciente)
- ksort(), krsort(): Ordena por claves un array (r=decreciente)
- array_combine(): Crea un array usando un array por claves y otro por sus valores
- array_walk_recursive(): Aplica una función de usuario recursivamente a cada miembro de un array

3. BIBLIOTECA DE FUNCIONES

Institut Joan d'Àustria

Funciones de fecha y hora

- date(): Formatea una fecha según un formato dado

  $fecha = date ("j/n/Y H:i");

  print ("$fecha"); → Resultado: 26/9/2005 17:36

- strtotime(): Convierte una fecha en un *timestamp* de UNIX

  $fecha = date ("j/n/Y", strtotime("5 april 2001"));

  print ("$fecha"); → Resultado: 5/4/2001

- idate(): Devuelve una fecha y hora como un entero
- date_sunset(): Devuelve la hora de la puesta de sol para un dia y localización
- date_sunrise(): Idem para la hora de salida del sol
- time_nanosleep(): Retarda un numero de nano segundos

3. BIBLIOTECA DE FUNCIONES

Símbolos de formato para Date()

‘M’	Jan
‘F’	January
‘m’	01
‘n’	1

Dia del Mes	‘d’	01
Dia del Mes	‘J’	1
Dia de la semana	‘l’	Monday
Día de la semana	‘D’	Mon

3. BIBLIOTECA DE FUNCIONES

Usos de Date()

```
$datedisplay = date(“yyyy/m/d”);
Print $datedisplay;
# Si la fecha es 1º de Abril del 2009
# mostrará  ‘2009/4/1’
$datedisplay = date(“l, F m, Y”);
Print $datedisplay;
# Si la fecha es 1º de Abril de 2009
# mostrará  ‘Miercoles, Abril 1, 2009’
```